

### REMARKS

Claims 1-19 are pending in the application of which claims 1, 9, and 17-19 are independent. By this amendment, Claims 1, 9, and 17-19 are amended. No new matter is introduced. All claims are believed to now be in condition for allowance.

As an initial matter, Applicant's attorneys wish to thank Examiner Truong for her participation in the telephonic interview of July 6, 2004. Applicant's attorneys note that the claim amendments herein, as well as the filing of this Request for Continued Examination are consistent with those discussions during interview.

Turning briefly to the Dundon reference, a dynamic Application Program Interface (API) mapping is described for architectures having multiple application environments. The API is described as "support[ing] linking and binding [of] the application 116 with the libraries 120 available" (Column 3, lines 44-45). Dundon describes a problem that can occur in these architectures when different libraries are available. Dundon refers to a client/server architecture as having a *runtime* environment and a *remote* environment, in which the libraries that contain *runtime* APIs are different from the libraries that contain *remote* APIs. Consequently, there may be several APIs that are not supported in the *remote* API set. (Column 1, lines 40-53). If the API is not supported in a particular environment the application will be unable to link and bind with the required libraries.

To be sure that an API is supported in a particular environment, Dundon uses a dynamic API mapping process. Dundon summarizes this process as being "accomplished by loading and linking the required libraries before any APIs are called by calling a special function provided for this purpose, before the application tries to call any . . . APIs." (Column 4, line 66 through Column 5, line 3). When the special function is called, "all of the libraries are loaded and all APIs in the library are linked. From this point forward, these linked APIs can be called by the application." (Column 5, lines 3-6). Thus, Dundon does not allow an application to call any APIs until all of the libraries are loaded and the APIs linked by using the dynamic mapping process before any APIs are called.

In the Applicant's invention, an already executing API receives from an application a request for functionality (i.e., a call) to be fulfilled by a first component. In response to the received request, the first component is queried for the requested functionality. Upon receiving

an indication that the requested functionality is not implemented by the first component, a search is performed to locate an augmentation component that implements the requested functionality. Once found, the augmentation component is loaded to fulfill the request for functionality, enabling the API to respond to the call.

#### Claim Rejections Under 35 U.S.C. §103

Claims 1-3, 5-11, and 13-19 are rejected under 35 U.S.C. §103(a) as being unpatentable over Dundon (U.S. Patent No. 6,253,257 B1) in view of Goldberg et al. (U.S. Patent No. 6,496,833 B1). This rejection is respectfully traversed and reconsideration is requested.

Applicant's invention solves the problem of an API being unable to respond to calls by an application for lack of functionality, by providing the API with the necessary functionality in response to the call. A patentable distinction of the Applicant's invention over Dundon, is that it provides the required functionality in response to the API receiving a request from an application. This clarification that the API is responding to a request from an application is now recited in amended independent claims 1, 9, and 17-19.

For further clarification, amended claim 1 also recites that the process occurs within an executing API. Execution of the API by a processor suggests that it is already loaded into memory and linked. Thus, Applicant's invention allows an executing API to add functionality, as required, in response to an application call. Accordingly, the functionality of the API can be tailored, as required, during the course of execution and retailored again, all without the need for a preliminary configuration step.

With respect to independent claims 1, 9, 17-19, the prima facie requirement for obviousness rejection is not met because Dundon at least does not describe or suggest providing an API with the required functionality in response to receiving a request from an application. In contradistinction, Dundon loads and links the required libraries before any APIs are called. In fact, Dundon does not even allow an application to call an API until *all* of the libraries are loaded. Goldberg et al. describes a query object generator tool and is relied upon for teaching querying. Goldberg et al. does not describe or suggest providing required functionality in response to an API receiving an application call and, therefore, fails to remedy this deficiency.

Further, it would not have been obvious to modify Dundon to provide an API with the required functionality in response to receiving a request from an application. Dundon first requires that *all* of the libraries are loaded before an API is called. Having all of the libraries loaded would ensure that the API had the required functionality before a call to it is attempted by an application. This suggests that Dundon anticipates a failure should an API be called before having all of its libraries loaded. Thus, there is no likelihood of success for modifying Dundon to provide the required functionality in response to the API receiving a request from an application as in the present invention.

Claims 2-3 and 5-8, depend either directly or indirectly from base claim 1. As they contain all of the limitations of the base claim and any intervening claims, they are also not obvious in view of the combination of Dundon and Goldberg et al. for at least the same reasons as in claim 1. Similarly, claims 10-11, and 13-16 depend either directly or indirectly from base claim 9. As they contain all of the limitations of the base claim and any intervening claims, they are also not obvious in view of the combination of Dundon and Goldberg et al. for at least the same reasons as in claim 9. Accordingly, the rejection of claims 1-3, 5-11, and 13-19 under §103 is believed to be overcome. Acceptance is respectfully requested.

Claims 4 and 12 are rejected under 35 U.S.C. §103(a) as being unpatentable over Dundon (U.S. Patent No. 6,253,257 B1) in view of Goldberg et al. (U.S. Patent No. 6,496,833 B1) in view of Ramos et al. (U.S. Patent No. 5,896,533). As claim 4 ultimately depends from claim 1 and claim 12 ultimately depends from claim 9, these claims contain all of the limitations of their respective independent claims and any intervening claims. Ramos fails to remedy the deficiency discussed above in relation to amended base claims 1 and 9. Accordingly, claims 4 and 12 are also not obvious in view of the combination of Dundon, Goldberg et al., and Ramos et al. for at least the same reasons discussed above. Accordingly, this rejection is respectfully traversed and reconsideration requested.

**CONCLUSION**

In view of the above amendments and remarks, it is believed that all pending claims (Claims 1-19) are in condition for allowance, and it is respectfully requested that the application be passed to issue. If the Examiner feels that a telephone conference would expedite prosecution of this case, the Examiner is invited to call the undersigned.

Respectfully submitted,

HAMILTON, BROOK, SMITH & REYNOLDS, P.C.

By   
Ralph Tremontozzi  
Registration No. 55,686  
Telephone: (978) 341-0036  
Facsimile: (978) 341-0136

Concord, MA 01742-9133

Dated: July 9, 2004